Security_Issues

ASE > Users > NIC KEREALA Application Group 3 > Samunnathi090621 > Security Issues

Exported by NIC KEREALA Application Group 3 - 6/10/2021 9:14 AM
Generated on 6/9/2021 12:31:06 PM

Help for this report

Table of Contents

Executive Summary

Issue Types Discovered

	Issue Type	Number of Issues
!	SQL Injection	8
!	Cross-Site Request Forgery	71
!	Database Error Pattern Found	11
!	Permanent Cookie Contains Sensitive Session Information	1
!	Web Application Source Code Disclosure Pattern Found	1
i	Possible Server Path Disclosure Pattern Found	1

Affected URLs/Files

	URL/File	Number of Issues
❗	http://10.162.2.54:81/kswcf/index.php/AdmInst/InstDetailsEntry	3
❗	http://10.162.2.54:81/kswcf/index.php/AdmSearch/ViewRegStudDetls	3
❗	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptDetlsApprove	3
❗	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptverifyDetails	3
❗	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EditRegtrStudDetls	3
❗	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetls	2
❗	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsC	2
❗	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsClr	3
⚠	http://10.162.2.54:81/kswcf/index.php/AdmInst/EditInstDetails	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionView	2
⚠	http://10.162.2.54:81/kswcf/index.php/AdmInst/UpdateInstitutionMaster	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InsCourseInset	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InstCourseLoad	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/bank_view	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/BankDelte	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/bankDetsEdit	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/BoardView	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/DistrCtDetsUpdte	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/EligBleDetsUpdte	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRt_baNsk	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNerDetsUpdte	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNrDets	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtbaNskDetsUpdte	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoardDetsUpdte	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoRdDets	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtDistrCtDets	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtEligBleDets	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtReliGnDets	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtStateDets	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetsEdit	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetSView	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationDetsUpdte	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationEdit	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationView	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsEdit	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsUpdte	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetSView	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsEdit	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsUpdte	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateView	1
⚠	http://10.162.2.54:81/kswcf/index.php/AdmMasters/upDat_baNsk	1

!	http://10.162.2.54:81/kswcf/index.php/AdmMasters/view_bank_afteredit	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/CourseGroupCreate	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/CourseLoad	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/coursemast_view	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_insert	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_view	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/delete_coursetype	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/EditCourseMaster	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/EntryCourseMast	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/InsCourseMaster	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/new_entry	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/update_coursetype	1
!	http://10.162.2.54:81/kswcf/index.php/Admn/view_coursetype_afteredit	1
!	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedCancelList	1
!	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedGenList	1
!	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProcesSeltedStudList	1
!	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetsAlotmentStudList	1
!	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/studViewProceedings	1
!	http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchBankDets	1
!	http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchInstView	1
!	http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchStudView	1
!	http://10.162.2.54:81/kswcf/index.php/AdmStatistics/CoursStaticsAdmFresh	1
!	http://10.162.2.54:81/kswcf/index.php/AdmStatistics/InstStatVew	1
!	http://10.162.2.54:81/kswcf/index.php/AdmStatistics/StudListView	1
!	http://10.162.2.54:81/kswcf/index.php/AdmStatistics/studViewReg	1
!	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegChangeDets	1
!	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/RegStudDetlsCancel	1
!	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/StudregpageEdit	1
!	http://10.162.2.54:81/kswcf/index.php/main/captcha	1
!	http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetInstionSMSmail	1
!	http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetstatusSMSmail	1
!	http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/SendIndualsmsact	1
⚠	http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets	3
⚠	http://10.162.2.54:81/kswcf/index.php/Admn/AdmnFrntPge/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b946f	1
ℹ	http://10.162.2.54:81/kswcf/jscript/getdist.php	1

Fix Recommendations

	Fix Recommendations	Number of Affected Issues
❗	Review possible solutions for hazardous character injection	19
⚠	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	71
◆	Avoid storing sensitive session information in permanent cookies	1
◆	Remove source code files from your web-server and apply any relevant patches	1
ℹ	Download the relevant security patch for your web server or web application.	1

Security Risks

	Risk	Number of Issues
❗	It is possible to view, modify or delete database entries and tables	19
🛡	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user	71
⚠	It is possible to retrieve the source code of server-side scripts, which may expose the application logic and other sensitive information such as usernames and passwords	1
⚠	It may be possible to steal session information (cookies) that was kept on disk as permanent cookies	1
ℹ	It is possible to retrieve the absolute path of the web server installation, which might help an attacker to develop further attacks and to gain information about the file system structure of the web application	1

WASC Threat Classification

WASC Threat Classification	Number of Issues
Authorization: Insufficient Session Expiration	1
Command Execution: SQL Injection	19
Cross-site Request Forgery	71
Information Disclosure: Information Leakage	2

Issues Sorted by Severity

There are 93 issues of 6 different types across 76 URLs

[High] http://10.162.2.54:81/kswcf/index.php/AdmInst/InstDetailsEntry - 3 issue(s)

Issue 1 of 3

[High] SQL Injection

Issue:	4580749
Severity:	High
URL:	http://10.162.2.54:81/kswcf/index.php/AdmInst/InstDetailsEntry
Parameter:	insttype_id
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 4

The following changes were applied to the original request:

```
Set the value of the parameter 'insttype_id' to '4%27%3B'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmInst/InstDetailsEntry]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmInst/InstDetailsEntry HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionMaster/8fdb7d1450c595cc4ff6ce7676df7d4f3f
2b946f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

insttype id=4%27%3B
&insttype name=RHGHGFH&inststate=18&instdist=1&street=hgfhgfh&city=gfhgfh&post_office=hgfhgf&pin_c
ode=654654&ownde_by=G&dceinst=Y

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1542
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:42:20 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:42:20 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
```

```
atabaseError Occurred</h1> <p>Error Number: 1064</p><p> You have an error in your SQL
syntax ; check the manual that corresponds to your MariaDB server version for the right syntax to
use near 'amp amp 0390000 AND 50000' at line 1</p><p>select max(inst code)+1 as maxinstcode from
institution details where inst_type='4 amp amp 039' and inst_code BETWEEN 4 amp amp 0390000 AND
50000 </p><p
...
<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 1064</p><p> You have an error in your SQL
syntax ; check the manual that corresponds to your MariaDB server version for the right syntax to
use near 'amp amp 0390000 AND 50000' at line 1</p><p>select max(inst code)+1 as maxinstcode from
institution details where inst_type='4 amp amp 039' and inst_code BETWEEN 4 amp amp 0390000 AND
50000 </p><p
...
```

Issue 2 of 3

[Medium] Cross-Site Request Forgery

Issue:	4580790
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmInst/InstDetailsEntry
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmInst/InstDetailsEntry]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmInst/InstDetailsEntry HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

insttype id=4&insttype name=RHGHGFH&inststate=18&instdist=1&street=hgfhgfh&city=gfhgfh&post_office
=hgfhgf&pin_code=654654&ownde_by=G&dceinst=Y

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 75
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=94
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:09 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

40313|Institution is registered Successfully Institution Code is :40313|1
```

Issue 3 of 3

[Low] Database Error Pattern Found

Issue:	4580805
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/AdmInst/InstDetailsEntry
Parameter:	insttype_id
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'insttype_id' to '4WFXSSProbe'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmInst/InstDetailsEntry]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmInst/InstDetailsEntry HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionMaster/8fdb7d1450c595cc4ff6ce7676df7d4f3f
2b946f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

insttype id=4WFXSSProbe
&insttype name=RHGHGFH&inststate=18&instdist=1&street=hgfhgfh&city=gfhgfh&post_office=hgfhgf&pin_c
ode=654654&ownde_by=G&dceinst=Y

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1412
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:42:17 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:42:17 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabase Error Occurred</h1> <p>Error Number: 1054</p><p>Unknown column '4WFXSSProbe0000'
in 'where clause'</p><p>select max(inst code)+1 as maxinstcode from institution details where
inst type='4WFXSSProbe' and inst code BETWEEN 4WFXSSProbe0000 AND 50000 </p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/
...
<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A
DatabaseError Occurred</h1> <p>Error Number: 1054</p><p>Unknown column '4WFXSSProbe0000'
in 'where clause'</p><p>select max(inst code)+1 as maxinstcode from institution details where
inst type='4WFXSSProbe' and inst code BETWEEN 4WFXSSProbe0000 AND 50000 </p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/
...
```

[High] http://10.162.2.54:81/kswcf/index.php/AdmSearch/ViewRegStudDetls - 3 issue(s)

Issue 1 of 3

[High] SQL Injection

Issue:	4580718
Severity:	High
URL:	http://10.162.2.54:81/kswcf/index.php/AdmSearch/ViewRegStudDetls
Parameter:	rentype
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'rentype' to 'F"'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmSearch/ViewRegStudDetls]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmSearch/ViewRegStudDetls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmSearch/StudAdmSerch/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b9
46f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

reg_no=010045354317& rentype=F" &ssyear=2020

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:43:46 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:43:46 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

Issue 2 of 3

[Medium] Cross-Site Request Forgery

Issue:	4580801
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmSearch/ViewRegStudDetls
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

The test result seems to indicate a vulnerability because the Test Response is identical to the Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmSearch/ViewRegStudDetls]]>

Request/Response:

```
POST /kswcf/index.php/AdmSearch/ViewRegStudDetls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

reg_no=010045354317&rentype=F&ssyear=2020

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=70
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:16 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8



<table align="center" class="table table-hover">
<tr ><td align="right"><b><font color="#0000FF">Registered Date and Time:&nbsp;
</font></b><b><font color="#FF0000">2021-06-09 06:12:24</font></b></td>
</tr></table>





<table width="100%" height="473" border="1" align="left" valign="top" style="margin-
left:0px;margin-top:0px;" class="table table-hover">
<tr class="view1">
<td width="53%" height="23" align="left"><span class="style14">Student WRID </span></td>
<td width="47%" ><strong>010045354317</strong></td>

...
```

Issue 3 of 3

[Low] Database Error Pattern Found

Issue:	4580767
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/AdmSearch/ViewRegStudDetls
Parameter:	rentype
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'rentype' to 'FWFXSSProbe'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmSearch/ViewRegStudDetls]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmSearch/ViewRegStudDetls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmSearch/StudAdmSerch/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b9
46f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

reg_no=010045354317& rentype=FWFXSSProbe &ssyear=2020

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:43:42 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:43:42 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabase Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabase Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
```

```
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptDetlsApprove - 3 issue(s)

Issue 1 of 3

[High] SQL Injection

Issue:	4580773
Severity:	High
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptDetlsApprove
Parameter:	freshren
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'freshren' to 'F"'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptDetlsApprove]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/DeptDetlsApprove HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/deptVerify/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b9
46f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ssyear=2020&sstype=DRAPM& freshren=F" &studid=

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:44:19 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:44:19 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A Database Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

Issue 2 of 3

[Medium] Cross-Site Request Forgery

Issue:	4580760
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptDetlsApprove
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptDetlsApprove]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/DeptDetlsApprove HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ssyear=2020&sstype=DRAPM&freshren=F&studid=

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 2078
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=82
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:13 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table id="example" class="table table-hover">
<thead>
<tr bgcolor="#E1FFF0">
<td colspan="13" align="center"><font color="#0000CC" size="3"><b>Approval Student
List</b></font></td>
</tr>
<tr >
<th align="center"><b>SI.No</b></th>
<th align="center"><b>Scholarship Status</b></th>
<th align="center"><b>Register No</b></th>
<th align="center"><b>Scholarship Year</b></th>
<th align="center"><b>Name</b></th>
<th align="center"><b>Institution Name</b></th>
<th align="center"><b>Course Name</b></th>
<th align="center"><b
...
```

Issue 3 of 3

[Low] Database Error Pattern Found

Issue:	4580752
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptDetlsApprove
Parameter:	freshren
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'freshren' to 'FWFXSSProbe'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptDetlsApprove]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/DeptDetlsApprove HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/deptVerify/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b9
46f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ssyear=2020&sstype=DRAPM& freshren=FWFXSSProbe &studid=

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:44:15 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:44:15 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabase Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabase Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
```

```
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptverifyDetails - 3 issue(s)

Issue 1 of 3

[High] SQL Injection

Issue:	4580745
Severity:	High
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptverifyDetails
Parameter:	freshren
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'freshren' to 'F"'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptverifyDetails]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/DeptverifyDetails HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/deptVerify/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b9
46f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

reg_no=010033380112&sstype=DRAPM&ssyear=2020& freshren=F"

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:44:21 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:44:21 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A
Database Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A Database Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

Issue 2 of 3

[Medium] Cross-Site Request Forgery

Issue:	4580755
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptverifyDetails
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptverifyDetails]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/DeptverifyDetails HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

reg_no=010033380112&sstype=DRAPM&ssyear=2020&freshren=F

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=77
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:14 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="row">
<div class="col-md-4">

<div class="row">
<div class="col-md-12">
</div>
</div>
</div>
</div>

<form role="form" id="verifydet"
action="http://10.162.2.54:81/kswcf/index.php/AdmStudDets/StudVeryInstSave" method="post"
autocomplete="OFF" >
<input type="hidden" name="regval" class="form-control required" value="01003338011
...
```

Issue 3 of 3

[Low] Database Error Pattern Found

Issue:	4580761
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptverifyDetails
Parameter:	freshren
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'freshren' to 'FWFXSSProbe'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptverifyDetails]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/DeptverifyDetails HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/deptVerify/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b9
46f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

reg_no=010033380112&sstype=DRAPM&ssyear=2020& freshren=FWFXSSProbe

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:44:16 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:44:16 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
```

```
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EditRegtrStudDetls - 3 issue(s)

Issue 1 of 3

[High] SQL Injection

Issue:	4580727
Severity:	High
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EditRegtrStudDetls
Parameter:	freshren
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'freshren' to 'F"'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EditRegtrStudDetls]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/EditRegtrStudDetls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EdtstudDet/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b9
46f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ssyear=2020&sstype=DRAPM& freshren=F" &studid=

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:44:22 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:44:21 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A Database Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A Database Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

Issue 2 of 3

[Medium] Cross-Site Request Forgery

Issue:	4580738
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EditRegtrStudDetls
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EditRegtrStudDetls]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/EditRegtrStudDetls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ssyear=2020&sstype=DRAPM&freshren=F&studid=

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 5547
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=79
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:14 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table id="example" class="table table-hover">
<thead>
<tr bgcolor="#E1FFF0">
<td colspan="13" align="center"><font color="#0000CC" size="3"><b>Registered Student
List</b></font></td>
</tr>
<tr >
<th align="center"><b>SI.No</b></th>
<th align="center"><b>Scholarship Status</b></th>
<th align="center"><b>Register No</b></th>
<th align="center"><b>Scholarship Year</b></th>
<th align="center"><b>Name</b></th>
<th align="center"><b>Institution Name</b></th>
<th align="center"><b>Course Name</b></th>
<th align="center">
...
```

Issue 3 of 3

[Low] Database Error Pattern Found

Issue:	4580758
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EditRegtrStudDetls
Parameter:	freshren
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'freshren' to 'FWFXSSProbe'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EditRegtrStudDetls]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/EditRegtrStudDetls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EdtstudDet/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b9
46f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ssyear=2020&sstype=DRAPM& freshren=FWFXSSProbe &studid=

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:44:18 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:44:18 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A
Database Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A
Database Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
```

```
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetls - 2 issue(s)

Issue 1 of 2

[High] SQL Injection

Issue:	4580751
Severity:	High
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetls
Parameter:	fresh
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'fresh' to 'F"'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetls]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/InstRegStudDetls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/hec_vwstd/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b94
6f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

regno=010033380112& fresh=F" &ssyear=2020

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:42:53 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:42:52 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A Database Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

Issue 2 of 2

[Low] Database Error Pattern Found

Issue:	4580783
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetls
Parameter:	fresh
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'fresh' to 'FWFXSSProbe'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetls]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/InstRegStudDetls HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/hec_vwstd/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b94
6f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

regno=010033380112& fresh=FWFXSSProbe &ssyear=2020

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:42:47 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:42:47 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
```

```
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsC - 2 issue(s)

Issue 1 of 2

[High] SQL Injection

Issue:	4580797
Severity:	High
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsC
Parameter:	fresh
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'fresh' to 'F"'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsC]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/InstRegStudDetlsC HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/ChanDelDupID/8fdb7d1450c595cc4ff6ce7676df7d4f3f2
b946f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

regno=010033380113& fresh=F" &ssyear=2020

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:43:05 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:43:05 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A DatabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

Issue 2 of 2

[Low] Database Error Pattern Found

Issue:	4580746
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsC
Parameter:	fresh
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'fresh' to 'FWFXSSProbe'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsC]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/InstRegStudDetlsC HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/ChanDelDupID/8fdb7d1450c595cc4ff6ce7676df7d4f3f2
b946f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

regno=010033380113& fresh=FWFXSSProbe &ssyear=2020

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:43:01 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:43:01 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
```

```
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsClr - 3 issue(s)

Issue 1 of 3

[High] SQL Injection

Issue:	4580796
Severity:	High
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsClr
Parameter:	fresh
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'fresh' to 'F"'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsClr]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/InstRegStudDetlsClr HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/clrregid/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b946
f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

regno=010033380112& fresh=F" &ssyear=2020

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:44:07 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:44:07 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabase Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

Issue 2 of 3

[Medium] Cross-Site Request Forgery

Issue:	4580759
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsClr
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsClr]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/InstRegStudDetlsClr HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

regno=010033380112&fresh=F&ssyear=2020

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=82
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:13 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8



<table align="center" class="table table-hover">
<tr ><td align="right"><b><font color="#0000FF">Registered Date and Time:&nbsp;
</font></b><b><font color="#FF0000">2021-04-20 09:44:24</font></b></td>
</tr></table>




<table width="100%" height="473" border="1" align="left" valign="top" style="margin-
left:0px;margin-top:0px;" class="table table-hover">
<tr class="view1">
<td width="53%" height="23" align="left"><span class="style14">Student WRID </span></td>
<td width="47%" ><strong>010033380112</strong></td>

...
```

Issue 3 of 3

[Low] Database Error Pattern Found

Issue:	4580770
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsClr
Parameter:	fresh
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
Set the value of the parameter 'fresh' to 'FWFXSSProbe'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsClr]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/InstRegStudDetlsClr HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmStudDets/clrregid/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b946
f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

regno=010033380112& fresh=FWFXSSProbe &ssyear=2020

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1216
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=8k4pk1cm8utt39cu0tuiple6fqdn47m8; expires=Wed, 09-Jun-2021 08:44:03 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:44:02 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabase Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
C:/xampp/htdocs/kswcf/system/database/DB driver.php</p><p>Line Number: 691</p> </div> </body>
</html><!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error
</title> <style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-
selection { background-color: #E13300; color: white; } body { background-color: #fff; margin:
40px; font: 13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399;
bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabase Error Occurred</h1> <p>Error Number: 0</p><p></p><p></p><p>Filename:
```

```
C:/xampp/htdocs/kswcf/system/database/DB_driver.php</p><p>Line Number: 691</p> </div> </body>
</html>
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInst/EditInstDetails - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580793
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmInst/EditInstDetails
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmInst/EditInstDetails]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmInst/EditInstDetails HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

insttype=4&state1=0&dist1=0

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 340
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=85
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:09 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table bgcolor='#DEE6D9' width='100%' style='margin-left:10px;' class='table table-hover'>
<tr class='trset' bgcolor='#DEE6D9'>
<td ><b>Sl no</b></font></td>
<td ><b>Institution Code</b></td>
<td ><b>Institution Name</b></td>
<td ><b>Edit</b></td>
<!--<td ><b>Delete</b></td>-->
</tr></table>
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionView - 2 issue(s)

Issue 1 of 2

[Medium] Cross-Site Request Forgery

Issue:	4580735
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionView
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmInst/InstitutionView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

insttype=0&state=0&dist=0

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 241
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=86
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:08 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table class='table table-hover' borser='1'>
<tr class='trset' bgcolor='#DEE6D9'>
<td ><b>Sl No</b></td>
<td><b>Institution Code</b></td>
<td ><b>Institution Name</b></td>
<td ><b>District</b></td>
</tr></table>
```

Issue 2 of 2

[Low] Permanent Cookie Contains Sensitive Session Information

Issue:	4580804
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionView
Response Cookie:	ci_session
Risk(s):	It may be possible to steal session information (cookies) that was kept on disk as permanent cookies
Fix:	Avoid storing sensitive session information in permanent cookies

Variant 1 of 1

Reasoning:

```
AppScan found that a session id cookie is stored on the client machine.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmInst/InstitutionView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer:
http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionMaster/8fdb7d1450c595cc4ff6ce7676df7d4f3f
2b946f
Connection: keep-alive
Host: 10.162.2.54:81
X-Requested-With: XMLHttpRequest
Accept: */*
Origin: http://10.162.2.54:81
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

insttype=0&state=0&dist=0

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 241
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=99
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=fhbsfobvf46s5gtarc0ikshbq2ftr1jl; expires=Wed, 09-Jun-2021 08:02:31 GMT;
Max-Age=7200; path=/; HttpOnly
Date: Wed, 09 Jun 2021 06:02:31 GMT
Referrer-Policy: same-origin
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table class='table table-hover' borser='1'>
<tr class='trset' bgcolor='#DEE6D9'>
<td ><b>Sl No</b></td>
<td><b>Institution Code</b></td>
<td ><b>Institution Name</b></td>
<td ><b>District</b></td>
</tr></table>
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInst/UpdateInstitutionMaster - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580792
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmInst/UpdateInstitutionMaster
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmInst/UpdateInstitutionMaster]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmInst/UpdateInstitutionMaster HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

insttype=4&instcode=40311&instname=RHGHGFH&inststate=18&instdist=Thiruvananthapuram&owned=G&dceins
t2=Y

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InsCourseInset - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580768
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InsCourseInset
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InsCourseInset]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmInstCourse/InsCourseInset HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ctypeview=UG&instutype=4&instname=40311&coursecode%5B%5D=3&coursecode%5B%5D=4&coursecode%5B%5D=6&c
oursecode%5B%5D=8&coursecode%5B%5D=509&coursecode%5B%5D=510&aided%5B%5D=Y&aided%5B%5D=Y&aided%5B%5
D=Y&aided%5B%5D=Y&aided%5B%5D=Y&aided%5B%5D=Y

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InstCourseLoad - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580802
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InstCourseLoad
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 2

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InstCourseLoad]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmInstCourse/InstCourseLoad HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ctypeview=0&instutype=0&instname=0

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=93
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:07 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

3
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/bank_view - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580766
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/bank_view
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/bank_view]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/bank view HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=72
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:20 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example' class='table table-hover' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='5' align='center'><font color='#0000CC' size='3'><b>View Bank Details</b></font></td>
</tr>
<tr>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>Bank
Code</b></font></td>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>Bank
Name</b></font></td>
<td style='font-size:12px' align='center'><fo
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/BankDelte - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580764
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/BankDelte
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/BankDelte]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/BankDelte HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

bankcd=86&bank=FGFDGFD&babbr=GFDGFDGFDG

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/bankDetsEdit - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580747
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/bankDetsEdit
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/bankDetsEdit]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/bankDetsEdit HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

banktype=6&bankstate=18&bankdist=1

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=56
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:23 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example4' class='table table-hover' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='5' align='center'><font color='#0000CC' size='3'><b>Edit Bank Details</b></font></td>
</tr>
<tr>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-
serif'><b>SI.NO</b></font></td>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>Branch
Code</b></font></td>
<td style='font-size:12px' align='center'><fon
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/BoardView - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580748
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/BoardView
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/BoardView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/BoardView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

bords=1

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 7902
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=52
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:23 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example' class='table table-hover' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='5' align='center'><font color='#0000CC' size='3'><b>View Board
Details</b></font></td>
</tr>
<tr >
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>Board
Code</b></font></td>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>Board
Name</b></font></td>
<td style='font-size:12px' align='center'><fon
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/DistrCtDetsUpdte - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580771
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/DistrCtDetsUpdte
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/DistrCtDetsUpdte]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/DistrCtDetsUpdte HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

stname=Kerala&stcode=18&distnam=TYRTYTRYR+GFGFHGFHFG&distcode=15

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/EligBleDetsUpdte - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580778
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/EligBleDetsUpdte
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/EligBleDetsUpdte]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/EligBleDetsUpdte HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

sstype=DRAPM&heads=sadsadsa&descr=dsadsadsadsad+ewefrewrwer+ewrewrewrewrewr+werew+er+rewrwerwe&dis
order=3&slno=178

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRt_baNsk - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580753
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRt_baNsk
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRt_baNsk]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRt baNsk HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8


bankname=FGFDGFD&bankabbr=GFDGFDGFDG


HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8


1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNerDetsUpdte - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580782
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNerDetsUpdte
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNerDetsUpdte]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRtBanNerDetsUpdte HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

banner_id=915&banner_text=dfgfd+g+fdg+fdg+fdg+dfgfd+fsdfdsgdsg&status=E

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNrDets - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580737
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNrDets
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNrDets]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRtBanNrDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

banner_text=dfgfd+g+fdg+fdg+fdg+dfgfd&status=E

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtbaNskDetsUpdte - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580775
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtbaNskDetsUpdte
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtbaNskDetsUpdte]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRtbaNskDetsUpdte HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8


bank code=6&b code=08525&b name=ATTINGAL&ifsc code=BKID0008525&b address=ATTINGAL+P.B.NO.9++ATTING
AL++DIST.THIRUVANANTHAPURAM++TRIVANDRUM+695101+KERALA&inststate=18&instdist=1&pin_code=695101&txtE
mail=


HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8


1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoardDetsUpdte - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580807
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoardDetsUpdte
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoardDetsUpdte]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRtBoardDetsUpdte HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

board=FDGFDGFD+GFDGFG&babbr=GFDGFGFDGF&boardcd=19

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoRdDets - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580736
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoRdDets
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoRdDets]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRtBoRdDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

boardname=FDGFDGFD&boardabbr=GFDGFGFDGF

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtDistrCtDets - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580741
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtDistrCtDets
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtDistrCtDets]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRtDistrCtDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

statename=18&distname=TYRTYTRYR

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtEligBleDets - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580723
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtEligBleDets
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtEligBleDets]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRtEligBleDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

sstype=DRAPM&head=sadsadsa&desc=dsadsadsadsad&order=3

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtReliGnDets - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580774
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtReliGnDets
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtReliGnDets]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRtReliGnDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

religion=fdsfdsf+dfds+f+sdfs

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtStateDets - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580788
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtStateDets
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtStateDets]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/InsRtStateDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

statename=DFGFDGFDGD

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetsEdit - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580789
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetsEdit
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetsEdit]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/NewsUpdtDetsEdit HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

flg=1

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 5836
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=58
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:25 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example4' class='table table-hover' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='7' align='center'><font color='#0000CC' size='3'><b>Edit News and Updates
Details</b></font></td>
</tr>
<tr>
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Slno</b></font></td>
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Date</b></font></td>
<td style='f
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetSView - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580765
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetSView
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetSView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/NewsUpdtDetSView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

flg=1

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 3958
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=59
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:25 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example' class='table table-hover' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='7' align='center'><font color='#0000CC' size='3'><b>View News and Update
Details</b></font></td>
</tr>
<tr >
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Slno</b></font></td>
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Date</b></font></td>
<td style='fon
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationDetsUpdte - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580795
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationDetsUpdte
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationDetsUpdte]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/NotificationDetsUpdte HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

news_text=test+abc+aaa+fdfdgfd&slno=842&popupDatepicker1_up=09%2F06%2F2021&status=Y&sstype=DRAPM

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationEdit - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580729
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationEdit
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationEdit]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/NotificationEdit HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8


flg=1


HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 6111
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=59
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:24 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8


<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example4' class='table table-hover' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='8' align='center'><font color='#0000CC' size='3'><b>Edit Notification
Details</b></font></td>
</tr>
<tr class='trset'>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-
serif'><b>Slno</b></font></td>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>Scholarship
Type</b></font></td>
<td style='font-size:
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationView - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580780
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationView
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/NotificationView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8


flg=1


HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 3772
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=61
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:24 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8


<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example' class='table table-hover' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='6' align='center'><font color='#0000CC' size='3'><b>View Notification
Details</b></font></td>
</tr>
<tr class='trset'>
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Slno</b></font></td>
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Scholarship Type</b></font></td>
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsEdit - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580754
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsEdit
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsEdit]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/ReliGnDetsEdit HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

flg=N

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=36
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:30 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example4' class='table' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='6' align='center'><font color='#0000CC' size='3'><b>Edit Religion
Details</b></font></td>
</tr>
<tr class='trset'>
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Sl no</b></font></td>
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Religion</b></font></td>
<td width='50
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsUpdte - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580763
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsUpdte
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsUpdte]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/ReliGnDetsUpdte HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

religion=fdsfdsf+dfds+f+sdfs+r+ewrew+rwe&religioncode=97

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetSView - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580722
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetSView
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetSView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/ReliGnDetSView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8


flg=N


HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=47
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:30 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8


<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example' class='table' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='5' align='center'><font color='#0000CC' size='3'><b>View Religion
Details</b></font></td>
</tr>
<tr>
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Sl no</b></font></td>
<td style='font-size:12px' align='center'><font color='#660000' face='Arial, Helvetica, sans-
serif'><b>Religion</b></font></td>
</tr>
</thead>
<tbody><t
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsEdit - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580779
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsEdit
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsEdit]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/StateDetsEdit HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

flg=N

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=54
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:26 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example4' class='table' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='5' align='center'><font color='#0000CC' size='3'><b>Edit State
Details</b></font></td>
</tr>
<tr class='trset'>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>State
Code</b></font></td>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>State
Name</b></font></td>
<td width='50' style='font-size:12p
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsUpdte - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580724
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsUpdte
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsUpdte]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/StateDetsUpdte HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8


stname=DFGFDGFDGDGFHGFHGFH&stcode=43


HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8


1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateView - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580762
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateView
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/StateView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

flg=N

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=53
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:26 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example' class='table' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='5' align='center'><font color='#0000CC' size='3'><b>View State
Details</b></font></td>
</tr>
<tr>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>State
Code</b></font></td>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>State
Name</b></font></td>
</tr>
</thead>
<tbody><tr >
<td style='font-s
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/upDat_baNsk - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580742
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/upDat_baNsk
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/upDat_baNsk]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/upDat baNsk HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8


bank=FGFDGFD&babbr=GFDGFDGFDG&bankcd=86&acwidth=1


HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8


1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/view_bank_afteredit - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580743
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmMasters/view_bank_afteredit
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmMasters/view_bank_afteredit]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmMasters/view bank afteredit HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=75
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:20 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class='box-body table-responsive no-padding' style='background-color: #ffff;' >
<table id='example1' class='table table-hover' width='80%'>
<thead>
<tr bgcolor='#E1FFF0'>
<td colspan='7' align='center'><font color='#0000CC' size='3'><b>Edit Bank Details</b></font></td>
</tr>
<tr>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>Bank
Code</b></font></td>
<td style='font-size:12px' align='center'><font face='Arial, Helvetica, sans-serif'><b>Bank
Name</b></font></td>
<td style='font-size:12px' align='center'><f
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/CourseGroupCreate - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580772
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/CourseGroupCreate
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/CourseGroupCreate]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/CourseGroupCreate HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ctypeview=UG&group=D&coursecode%5B%5D=11&coursecode%5B%5D=12&coursecode%5B%5D=14&aided%5B%5D=Y&aid
ed%5B%5D=Y&aided%5B%5D=Y

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/CourseLoad - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580784
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/CourseLoad
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/CourseLoad]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/CourseLoad HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ctypeedt=UG

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=95
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:08 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8




<table class='table table-hover' borser='1'>
<tr class='trset' bgcolor='#DEE6D9'>
<td align="center" ><font ><strong>Course name</strong></font></td>
<td align="center" ><font ><strong>Aided/Unaided</strong></font></td>
<td align="center" ><font><strong><input type='checkbox' name='ad' id='ad'
onclick='checkall(this.value);'>Select all</strong></font></td>
</tr>
</tr><tr class='view2'> <td>B.A LIFE SCIENCE</td>
<td>Aided</td>
<td align="center">



<input type="checkbox" n
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/coursemast_view - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580732
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/coursemast_view
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/coursemast_view]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/coursemast view HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ctype=0

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=93
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:07 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

3
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_insert - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580719
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_insert
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_insert]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/coursetype insert HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

coursetype=TRYRTYTRYT&courseabbr=RYTRYTRYRT&courserank=0&processrank=0&yeardur=0&hfee=&mfee=&feeli
mit=

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_view - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580756
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_view
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_view]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/coursetype view HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 7561
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table id="examples" class="table table-hover">
<thead>
<tr bgcolor="#E1FFF0">
<td colspan="8" align="center"><font color="#0000CC" size="3"><b>View Course type</b></font></td>
</tr>
<tr>
<th ><b>Coursetype Name</b></th>
<th ><b>Coursetype Abbreviation</b></th>
<th ><b>Course hierarchy</b></th>
<th ><b>Process Rank</b></th>
<th ><b>Course Duration</b></th>
<th ><b>Hostel Fee</b></th>
<th ><b>Maintanance Fee</b></th>
<th ><b>Course Fee Limit</b></th>
</tr>
</thead>
<tbody><tr class='view2'>
<td >UNDER GRADUATE</td>
<td >UG</t
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/delete_coursetype - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580806
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/delete_coursetype
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/delete_coursetype]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/delete coursetype HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

coursetype=TRYRTYTRYT

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/EditCourseMaster - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580787
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/EditCourseMaster
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/EditCourseMaster]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/EditCourseMaster HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ctypeedt=0

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=93
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:07 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

3
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/EntryCourseMast - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580786
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/EntryCourseMast
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/EntryCourseMast]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/EntryCourseMast HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 5828
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=92
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:08 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

 <table class='table table-hover'>
 <tr>
 <td colspan='2' align='left'><font color='#0000CC' size='3'><b>Course Master Entry </b></td>
 </tr> <tr>
 <td align='left' style='font-size:12px'><font face='Arial, Helvetica, sans-serif'><b>Course
 Type</b></font</td>
 <td align='left'>
 <select name="coursetype" id="coursetype" maxlength="50" tabindex="1">
 <option value="0">Select</option><option value="MSCED"> SIX YEAR MSC.ED </option><option
 value="ADD">ASSOCIATE DEGREE ONE YEAR</option><option value="AD">ASSOCIATE DEGREE TWO
 YEARS</option><option value="AUDIT">AUDIT</option><option
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/InsCourseMaster - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580785
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/InsCourseMaster
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/InsCourseMaster]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/InsCourseMaster HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

course_name=TRYTRYTRY&courseabbr=TYTR&duration=6&coursetype=UG&sem=S&stream=1&aid=Y&Cgroup=D

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/new_entry - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580731
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/new_entry
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/new_entry]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/new entry HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 2806
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=96
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:04 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table class='table table-hover'>
<tr bgcolor='#E1FFF0'>
<td colspan='3' align='center'><font color='#0000CC' size='3'><b>Entry of Course
Type</b></font></td>
</tr> <tr>
<td >Course Type Name</td>
<td >
<input name="bankname" type="text" id="bankname" size="40" maxlength="50" tabindex="2"
onkeyup="text onkeyUp(event,this);" onkeypress="CheckName(event,this);" class="form-control
required"></td>
</tr>

<tr>
<td >Course Type Abbreviation</td>
<td>
<input type="text" autocomplete="off" name="bankabbr" id="bankabbr" size="40" maxlength="10" ta
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/update_coursetype - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580733
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/update_coursetype
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/update_coursetype]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/update coursetype HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

coursetype=TTTTTUGUGUU&courseabbr=TRYRTYTRYT&bankcd=1&courserank=0&processrank=0&yeardur=0&hfee=68
&mfee=76&feelimit=45

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:06 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

1
```

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/view_coursetype_afteredit - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580781
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/view_coursetype_afteredit
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/view_coursetype_afteredit]]>
```

Request/Response:

```
POST /kswcf/index.php/Admn/view coursetype afteredit HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:07 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table id="examples" class="table table-hover">
<thead>
<tr bgcolor="#E1FFF0">
<td colspan="10" align="center"><font color="#0000CC" size="3"><b>Edit Course type</b></font></td>
</tr>
<tr>
<th ><b>Coursetype Name</b></th>
<th ><b>Coursetype Abbreviation</b></th>
<th ><b>Course hierarchy</b></th>
<th ><b>Process Rank</b></th>
<th ><b>Course Duration</b></th>
<th ><b>Hostel Fee</b></th>
<th ><b>Maintanance Fee</b></th>
<th ><b>Course Fee Limit</b></th><td width='50' ><b>Edit</b></td>
<td width='50' ><b>Delete</b></td> </tr>
</thead>
<
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedCancelList - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580720
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedCancelList
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedCancelList]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmProcssSeltion/GetProceeedCancelList HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

sname=DRAPM&ss_year=2020&process_no=1&prdate=26%2F05%2F2021&prcdno=123-15KSWCF

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 137
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=73
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:18 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table>
<thead>
<tr>
<th colspan="5" align="center"><font color="red">No Data Found</font></th>
</tr>
</thead>
</table>
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedGenList - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580721
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedGenList
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedGenList]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmProcssSeltion/GetProceeedGenList HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

sname=DRAPM&ss_year=2020&process_no=1&prdate=26%2F05%2F2021&prcdno=123-15KSWCF

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 2478
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=64
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:20 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;">
<table class="table table-hover" width="100%">
<thead><tr bgcolor='#003366'><td align='center' colspan='11'><font color='#FFF' size='3'>View
Proceedings</font></td></tr><tr >
<th align='center' ><font ><strong>Sl.No.</strong></font></th><th align='center'><font
><strong>Select All<input type='checkbox' name='checkbox1' id='checkbox1' onclick='check();'
/></strong></font></th><th align='center' ><font><strong>Scholarship Type</strong></font></th><th
align='center' ><font ><strong>Scholarship Year<
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProcesSeltedStudList - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580739
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProcesSeltedStudList
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProcesSeltedStudList]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmProcssSeltion/GetProcesSeltedStudList HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

sname1=DRAPM&state1=18&district1=1&insttype1=4&instname1=40001&ss_year=2020&coursetype=UG&cur_cour
se_group=B&cur_course_name=3&process_no=1

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 137
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=73
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:18 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table>
<thead>
<tr>
<th colspan="5" align="center"><font color="red">No Data Found</font></th>
</tr>
</thead>
</table>
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetsAlotmentStudList - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580750
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetsAlotmentStudList
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetsAlotmentStudList]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmProcssSeltion/GetsAlotmentStudList HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

sname1=DRAPM&state1=18&district1=0&insttype1=0&instname1=0&ss_year=2020&coursetype=0&cur_course_gr
oup=0&cur_course_name=0

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 137
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=73
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:18 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table>
<thead>
<tr>
<th colspan="5" align="center"><font color="red">No Data Found</font></th>
</tr>
</thead>
</table>
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/studViewProceedings - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580726
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/studViewProceedings
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/studViewProceedings]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmProcssSeltion/studViewProceedings HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

ss_type=DRAPM&ss_year=2020&process_no=1&proceedings_ID=DRAPM20F0001

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1939
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=59
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:20 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;">
<table id="example2" class="table table-hover" width="100%">
<thead><tr><th align='center' ><font ><strong>Sl.No.</strong></font></th><th align='center'
><font><strong>RegNo</strong></font></th><th align='center' ><font
><strong>Name</strong></font></th><th align='center' ><font ><strong>Mobile
No</strong></font></th><th align='center' ><font><strong>Institution Name</strong></font></th><th
align='center' ><font><strong>Course</strong></font></th><th align='center' ><font
><strong>Percentage/ Grad
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchBankDets - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580799
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchBankDets
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchBankDets]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmSearch/searchBankDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

branchname=te&banktype=2

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=75
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:17 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;">
<table id="example" class="table table-hover" width="80%">
<thead>
<tr bgcolor="#E1FFF0">
<th colspan="13" align="center"><font color="#0000CC" size="3"><b>Bank Details</b></font></th>
</tr>
<tr class='trset'><td width='25' align='center' ><font ><strong>Sl.No.</strong></font></th><th
width='50' align='center' ><font><strong>Bank Type</strong></font></th><th width='75'
align='center' ><font><strong>Branch Name</strong></font></th><th width='125' align='center'
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchInstView - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580734
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchInstView
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchInstView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmSearch/searchInstView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

txtser=&insttype=4

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=78
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:15 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;" >
<table id="example" class="table table-hover" width="80%">
<thead>
<tr bgcolor="#E1FFF0">
<td colspan="13" align="center"><font color="#0000CC" size="3"><b>Institution
Details</b></font></td>
</tr>
<tr >
<th align="center"><b>Institution Type</b></th>
<th align="center"><b>Institution Code</b></th>
<th align="center"><b>Institution Name</b></th>
<th align="center"><b>State</b></th>
<th align="center"><b>District</b></th>
<th align="
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchStudView - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580728
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchStudView
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchStudView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmSearch/searchStudView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

rentype=F&txtser=te

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 810
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=78
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:15 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;">
<table id="example" class="table table-hover" width="80%">
<thead>
<tr bgcolor="#E1FFF0">
<td colspan="13" align="center"><font color="#0000CC" size="3"><b>Student Details</b></font></td>
</tr>
<tr >
<th align="center"><b>SI.No</b></th>
<th align="center"><b>Students Name</b></th>
<th align="center"><b>Institution Name</b></th>
<th align="center"><b>Annual Income</b></th>
</tr>
</thead>
<tbody><tr class=''>
<td>1</td>
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStatistics/CoursStaticsAdmFresh - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580730
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStatistics/CoursStaticsAdmFresh
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStatistics/CoursStaticsAdmFresh]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStatistics/CoursStaticsAdmFresh HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

sname1=DRAPM&ss_year=2020

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1012
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=28
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:32 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;">
<table id="example6" class="table table-hover" width="100%">
<thead><tr><th align='center' ><font><strong>Course Type</strong></font></th><th align='center'
><font ><strong>Applied</strong></font></th><th align='center' ><font
><strong>Verified</strong></font></th><th align='center' ><font
><strong>Approved</strong></font></th><th align='center' ><font
><strong>Processed</strong></font></th></tr></thead>
<tbody><tr class='view1'>
<td align='left' class='style4'>PG</td>
<td align='cent
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStatistics/InstStatVew - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580809
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStatistics/InstStatVew
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStatistics/InstStatVew]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStatistics/InstStatVew HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

state1=18&district1=0&insttype1=4&instname1=0

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=51
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:32 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;">
<table id="example3" class="table table-hover" width="100%">
<thead><tr><th align='center' ><font><strong>Sl no</strong></font></th><th align='center'
><font><strong>Institution Code</strong></font></th><th align='center' ><font><strong>Institution
Name</strong></font></th><th align='center' ><font><strong>Registered
students</strong></font></th><th align='center' ><font><strong>Verified</strong></font></th><th
align='center' ><font><strong>Approved</strong></font></th></tr>
</thead>
<tbody
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStatistics/StudListView - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580803
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStatistics/StudListView
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStatistics/StudListView]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStatistics/StudListView HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

sname1=DRAPM&state1=18&district1=0&insttype1=4&instname1=0&check1=2&ss_year=2020

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 3129
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=38
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:31 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;">
<table id="example2" class="table table-hover" width="100%">
<thead><tr><th align='center' ><font ><strong>Sl.No.</strong></font></th><th align='center'
><font><strong>RegNo</strong></font></th><th align='center' ><font
><strong>Name</strong></font></th><th align='center' ><font ><strong>Mobile
No</strong></font></th><th align='center' ><font><strong>Institution Name</strong></font></th><th
align='center' ><font><strong>Course</strong></font></th><th align='center' ><font
><strong>Percentage/ Grad
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStatistics/studViewReg - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580808
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStatistics/studViewReg
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStatistics/studViewReg]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStatistics/studViewReg HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

reg_no=010016810511&ssyear=2020&sstype=DRAPM

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1303
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=31
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:32 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table border="1" align="center" cellpadding="0" cellspacing="0" bordercolor="#00B8FE"
bgcolor="EEF5FF" class="table table-hover">

<tr><td align='left'><b>Register ID</b></td><td align='left'><b>010016810511</b></td></tr><tr><td
align='left'><b>Scholarship Type</b></td><td align='left'>Dr.Ambedkar Post-Matric
Scholarship</td></tr><tr><td align='left'><b>Scholarship Year</b></td><td
align='left'>2020</td></tr><tr><td align='left'><b>Mobile Number</b></td><td
align='left'>9495366978</td></tr><tr><td align='left'><b>Institution Name</b></td><td
align='left'>UNIVERSITY COLLEGE,
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegChangeDets - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580776
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegChangeDets
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegChangeDets]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/InstRegChangeDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

regno=010033380113&fresh=1

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 4924
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=77
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:12 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8



<table border="0" bgcolor="#FFFFFF" class="table table-hover">
<tr>
<td colspan="4"><font>a) Board of matriculation(10th) or equivalent exam passed</font><font
color='red'>*</font></td>
<td colspan="2" >
<select name="lstboard" id="lstboard" onchange="getid();" class="form-control" >
<option value="0">Select</option>
<option value='1' >
BOARD OF PUBLIC EXAMINATIONS KERALA STATE </option>
<option value='2' >
CENTRAL BOARD OF SECONDARY EDUCATION </option>

...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/RegStudDetlsCancel - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580810
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/RegStudDetlsCancel
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/RegStudDetlsCancel]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/RegStudDetlsCancel HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

regno=010033380112&fresh=F&ssyear=2020

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 2421
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=85
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:13 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<table id='studreg1' width='100%' align='center' border='0' cellpadding='0' class='table table-
hover' ><tr><td>Name</td><td colspan='3'><b>:DIVYA S R</b></td></tr><tr></tr><tr><td>Date of
Birth</td><td colspan='3'><b>:11/05/1997</b></td></tr><tr></tr><tr><td>Institution name</td><td
colspan='3'><b>:KKTM GOVT . COLLEGE PULLUT, THRISSUR</b></td></tr><tr></tr><tr><td>Scholasrship
Year</td><td colspan='3'><b>:2020</b></td></tr><tr></tr><tr bgcolor='#D9D9D9'><td align='center'
colspan='7'>
<font ><strong> List of registered scholarship</strong></font></td><tr bgcolor='#e6bae6'><td
align='ce
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/StudregpageEdit - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580769
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/AdmStudDets/StudregpageEdit
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/AdmStudDets/StudregpageEdit]]>
```

Request/Response:

```
POST /kswcf/index.php/AdmStudDets/StudregpageEdit HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

reg_no=010016810511&sstype=DRAPM&ssyear=2020

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=76
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:16 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<style>
.clear{clear:both;}
.nav nav-tabs
{
background-color: #f7f4f4;
border-bottom: 2px solid;
border-bottom-color : #e4e3e3;
}
.buttonreg {
border-radius: 4px;
background-color: #204D74;
border: none;
color: #FFFFFF;
text-align: center;
font-size: 15px;
padding: 10px;

transition: all 0.5s;
cursor: pointer;
margin: 5px;
```

```
        }

        .buttonreg span {
        cursor: pointer;
        display: inline-block;
        position: relative;
        transition: 0.5s;
        }

        .buttonreg span:after {
        content: '\00bb';
        position: absolute;
        opacity: 0;
        top: 0;
        right: -20px;

        ...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/main/captcha - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580725
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/main/captcha
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Removed the cookie 'ci_session'
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/main/captcha]]>
```

Request/Response:

```
POST /kswcf/index.php/main/captcha HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 126
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=100
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=sm1c2osenu3ob677bt858gat2h9vfk8i; expires=Wed, 09-Jun-2021 08:59:10 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:59:10 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html;charset=utf-8

<img src="http://10.162.2.54:81/kswcf/images/captcha/1623221950.2245.jpg" width="150" height="40"
style="border:0;" alt=" " />
```

[Medium] http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetInstionSMSmail - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580744
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetInstionSMSmail
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 2

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetInstionSMSmail]]>
```

Request/Response:

```
POST /kswcf/index.php/SendSMSMailDets/GetInstionSMSmail HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

state1=18&district1=1&insttype1=4

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=29
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:31 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;">
<table class="table table-hover" width="100%">
<thead><tr >
<th align='center' ><font ><strong>Sl.No.</strong></font></th><th align='center'
width='50'><font><strong>Sms All</strong></font>
<input type='checkbox' name='ad' id='ad' onclick='alla(this.value);'></th><th align='center'
><font><strong>Institution Name</strong></font></th><th align='center' ><font><strong>Mobile
Number</strong></font></th></tr>
</thead>
<tbody> <tr class="view2">
<td align='center' class='style4'>1</t
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetstatusSMSmail - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580800
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetstatusSMSmail
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetstatusSMSmail]]>
```

Request/Response:

```
POST /kswcf/index.php/SendSMSMailDets/GetstatusSMSmail HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

sname1=DRAPM&state1=18&district1=0&insttype1=0&instname1=0&check1=2&adhid1=A&ss_year=2020

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 3025
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=57
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:30 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<div class="box-body table-responsive no-padding" style="background-color: #ffff;">
<table class="table table-hover" width="100%">
<thead><tr >
<td align='center' ><font ><strong>Sl.No.</strong></font></td><td align='center'
width='20%'><font><strong>SMS/e-Mail All</strong></font><input type='checkbox' name='ad' id='ad'
onclick='alla(this.value);'></td><td align='center' ><font><strong>RegNo</strong></font></td><td
align='center' ><font ><strong>Name</strong></font></td><td align='center' ><font
><strong>Mobile</strong></font></td><td align='center' ><font><strong>Institution Na
...
```

[Medium] http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/SendIndualsmsact - 1 issue(s)

Issue 1 of 1

[Medium] Cross-Site Request Forgery

Issue:	4580757
Severity:	Medium
URL:	http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/SendIndualsmsact
Risk(s):	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user
Fix:	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form

Variant 1 of 1

The following changes were applied to the original request:

```
- Set header to 'http://bogus.referer.hcl.com'
- Removed the header 'X-Requested-With'
- Removed the header 'Origin'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the Test Response is identical to the
Original Response, indicating that the Cross-Site Request Forgery attempt was successful, even
though it included a fictive 'Referer' header.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/SendIndualsmsact]]>
```

Request/Response:

```
POST /kswcf/index.php/SendSMSMailDets/SendIndualsmsact HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://bogus.referer.hcl.com
Connection: keep-alive
Host: 10.162.2.54:81
Accept: */*
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded; charset=UTF-8

mobileno=4564564564&mobilecont=dfg455656456456456

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 22
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=30
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:31 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

SMS Successfully Send
```

[Low] http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets - 3 issue(s)

Issue 1 of 3

[Low] Database Error Pattern Found

Issue:	4580740
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
- Removed the cookie 'ci_session'
- Set path to '/kswcf/index.php/admn/AdmLogDets'
- Removed the cookie 'ci_session'
- Set path to '/kswcf/index.php/admn/AdmLogDets'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets]]>
```

Request/Response:

```
POST /kswcf/index.php/admn/AdmLogDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://10.162.2.54:81/kswcf/index.php/Admn
Connection: keep-alive
Host: 10.162.2.54:81
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Origin: http://10.162.2.54:81
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.9
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded

username=kswcf&passwd=9c544b1db69384388734b0c88ee16673

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1554
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci_session=c81ieji3sqteduvv7s6qf50rpad1fnr0; expires=Wed, 09-Jun-2021 08:59:08 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:59:08 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabaseError Occurred</h1> <p>Error Number: 1064</p><p> You have an error in your SQL
syntax ; check the manual that corresponds to your MariaDB server version for the right syntax to
use near 'PASSWORD))='9c544b1db69384388734b0c88ee16673' and dept='KSWCF'' at line 1</p><p>select *
from admin user where userid='kswcf' AND md5(concat(,PASSWORD))='9c544b1db69384388734b0c88ee16673'
and dept='K
```

```
...
<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A D
atabase Error Occurred</h1> <p>Error Number: 1064</p><p> You have an error in your SQL
syntax ; check the manual that corresponds to your MariaDB server version for the right syntax to
use near 'PASSWORD))='9c544b1db69384388734b0c88ee16673' and dept='KSWCF'' at line 1</p><p>select *
from admin user where userid='kswcf' AND md5(concat(,PASSWORD))='9c544b1db69384388734b0c88ee16673'
and dept='K
...
```

Issue 2 of 3

[Low] Database Error Pattern Found

Issue:	4580794
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets
Parameter:	passwd
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
- Removed the cookie 'ci_session'
- Set the value of the parameter 'passwd' to 'e207b4c6a79ebbab0d6d2ac3c649498d'
- Removed the cookie 'ci_session'
- Set the value of the parameter 'passwd' to 'Ae207b4c6a79ebbab0d6d2ac3c649498dB'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets]]>
```

Request/Response:

```
POST /kswcf/index.php/admn/AdmLogDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://10.162.2.54:81/kswcf/index.php/Admn
Connection: keep-alive
Host: 10.162.2.54:81
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Origin: http://10.162.2.54:81
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.9
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded

username=kswcf& passwd=e207b4c6a79ebbab0d6d2ac3c649498d

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1554
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=tuduj15oiv941bqtakc6k7g4mbcelpjj; expires=Wed, 09-Jun-2021 08:59:10 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:59:10 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A Database Error Occurred</h1> <p>Error Number: 1064</p><p> You have an error in your SQL
syntax ; check the manual that corresponds to your MariaDB server version for the right syntax to
use near 'PASSWORD))='e207b4c6a79ebbab0d6d2ac3c649498d' and dept='KSWCF'' at line 1</p><p>select *
from admin user where userid='kswcf' AND md5(concat(,PASSWORD))='e207b4c6a79ebbab0d6d2ac3c649498d'
and dept='K
...
...
ecure-Requests: 1
Cache-Control: max-age=0
```

```
Origin: http://10.162.2.54:81
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.9
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded

username=kswcf& passwd=Ae207b4c6a79ebbab0d6d2ac3c649498dB

HTTP/1.1 200 OK
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
X-Frame-Options: sameorigin
Set-Cookie: ci session=5e2b8ese5qokf0hnenksv6h7tuu6n07a; expires=Wed, 09-Jun-2021 08:59:10 GMT;
Max-Age=7200; path=/; HttpOnly
X-Content-Type-Options: nosniff
Expires: Thu, 19 Nov 1981 08:52:0
...
<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A Database Error Occurred</h1> <p>Error Number: 1064</p><p> You have an error in your SQL
syntax ; check the manual that corresponds to your MariaDB server version for the right syntax to
use near 'PASSWORD))='e207b4c6a79ebbab0d6d2ac3c649498d' and dept='KSWCF'' at line 1</p><p>select *
from admin user where userid='kswcf' AND md5(concat(,PASSWORD))='e207b4c6a79ebbab0d6d2ac3c649498d'
and dept='K
...
```

Issue 3 of 3

[Low] Database Error Pattern Found

Issue:	4580798
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets
Parameter:	username
Risk(s):	It is possible to view, modify or delete database entries and tables
Fix:	Review possible solutions for hazardous character injection

Variant 1 of 1

The following changes were applied to the original request:

```
- Removed the cookie 'ci_session'
- Set the value of the parameter 'username' to 'kswcfWFXSSProbe'
```

Reasoning:

```
The test result seems to indicate a vulnerability because the response contains SQL Server errors.
This suggests that the test managed to penetrate the application and reach the SQL query itself,
by injecting hazardous characters.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets]]>
```

Request/Response:

```
POST /kswcf/index.php/admn/AdmLogDets HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://10.162.2.54:81/kswcf/index.php/Admn
Connection: keep-alive
Host: 10.162.2.54:81
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Origin: http://10.162.2.54:81
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.9
Accept-Language: en-US
Content-Type: application/x-www-form-urlencoded

username=kswcfWFXSSProbe &passwd=e207b4c6a79ebbab0d6d2ac3c649498d

HTTP/1.1 500 Internal Server Error
Connection: close
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1564
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: ci session=nfhpfiqgl1bdd9r6evnsnd7i3sm7s7b2; expires=Wed, 09-Jun-2021 08:59:09 GMT;
Max-Age=7200; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:59:09 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A Database Error Occurred</h1> <p>Error Number: 1064</p><p> You have an error in your SQL
syntax ; check the manual that corresponds to your MariaDB server version for the right syntax to
use near 'PASSWORD))='e207b4c6a79ebbab0d6d2ac3c649498d' and dept='KSWCF'' at line 1</p><p>select *
from admin user where userid='kswcfWFXSSProbe' AND
md5(concat(,PASSWORD))='e207b4c6a79ebbab0d6d2ac3c649498d' a
...
<!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title> Database Error </title>
<style type="text/css"> ::selection { background-color: #E13300; color: white; } ::-moz-selection
{ background-color: #E13300; color: white; } body { background-color: #fff; margin: 40px; font:
13px/20px normal Helvetica, Arial, sans-serif; color: #4F5155; } a { color: #003399; bac
...
```

```
...
border: 1px solid #D0D0D0; color: #002166; display: block; margin: 14px 0 14px 0; padding: 12px
10px 12px 10px; } #container { margin: 10px; border: 1px solid #D0D0D0; box-shadow: 0 0 8px
#D0D0D0; } p { margin: 12px 15px 12px 15px; } </style> </head> <body> <div id="container"> <h1>A Database Error Occurred</h1> <p>Error Number: 1064</p><p> You have an error in your SQL
syntax ; check the manual that corresponds to your MariaDB server version for the right syntax to
use near 'PASSWORD))='e207b4c6a79ebbab0d6d2ac3c649498d' and dept='KSWCF'' at line 1</p><p>select *
from admin user where userid='kswcfWFXSSProbe' AND
md5(concat(,PASSWORD))='e207b4c6a79ebbab0d6d2ac3c649498d' a
...
```

[Low] http://10.162.2.54:81/kswcf/index.php/Admn/AdmnFrntPge/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b946f - 1 issue(s)

Issue 1 of 1

[Low] Web Application Source Code Disclosure Pattern Found

Issue:	4580777
Severity:	Low
URL:	http://10.162.2.54:81/kswcf/index.php/Admn/AdmnFrntPge/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b946f
Risk(s):	It is possible to retrieve the source code of server-side scripts, which may expose the application logic and other sensitive information such as usernames and passwords
Fix:	Remove source code files from your web-server and apply any relevant patches

Variant 1 of 1

The following changes were applied to the original request:

```
Set path to '/kswcf/index.php/Admn/AdmnFrntPge/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b946f'
```

Reasoning:

```
The response contains source code of script files, which may expose sensitive information about
the site and the application logic.
<![CDATA[http://10.162.2.54:81/kswcf/index.php/Admn/AdmnFrntPge/8fdb7d1450c595cc4ff6ce7676df7d4f3f
2b946f]]>
```

Request/Response:

```
GET /kswcf/index.php/Admn/AdmnFrntPge/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b946f HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets
Connection: Keep-Alive
Host: 10.162.2.54:81
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=97
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:41:08 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8


...
mall-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>-->
</div>
</div><!-- ./col -->

<!--<div class="col-lg-3 col-xs-6">
<div class="small-box bg-red">
<div class="inner">
<h3> <?php // echo $reg->trans;?></h3>
<p>Total Transferred Amount</p>
</div>
<div class="icon">
<i class="ion ion-briefcase"></i>
</div>
</div>
</div>-->




...
...
mall-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>-->
```

```
         </div>
         </div><!-- ./col -->

         <!--<div class="col-lg-3 col-xs-6">
         <div class="small-box bg-red">
         <div class="inner">
         <h3> <?php // echo $reg->trans;?></h3>
         <p>Total Transferred Amount</p>
         </div>
         <div class="icon">
         <i class="ion ion-briefcase"></i>
         </div>
         </div>
         </div>-->




         ...
```

[Information] http://10.162.2.54:81/kswcf/jscript/getdist.php - 1 issue(s)

Issue 1 of 1

[Information] Possible Server Path Disclosure Pattern Found

Issue:	4580791
Severity:	Information
URL:	http://10.162.2.54:81/kswcf/jscript/getdist.php
Risk(s):	It is possible to retrieve the absolute path of the web server installation, which might help an attacker to develop further attacks and to gain information about the file system structure of the web application
Fix:	Download the relevant security patch for your web server or web application.

Variant 1 of 1

The following changes were applied to the original request:

```
- Set path to '/kswcf/jscript/getdist.php'
- Set query string to 'status=12'
```

Reasoning:

```
The response contains the absolute paths and/or filenames of files on the server.
<![CDATA[http://10.162.2.54:81/kswcf/jscript/getdist.php?status=12]]>
```

Request/Response:

```
GET /kswcf/jscript/getdist.php ? status=12 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 9.0; Win32)
Referer: http://10.162.2.54:81/kswcf/jscript/ajaxcode.js
Connection: Keep-Alive
Host: 10.162.2.54:81
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US

HTTP/1.1 200 OK
Connection: Keep-Alive
X-XSS-Protection: 1; mode=block
Server: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Pragma: no-cache
Content-Length: 1006
X-Frame-Options: sameorigin
X-Content-Type-Options: nosniff
Keep-Alive: timeout=5, max=96
Cache-Control: no-store, no-cache, must-revalidate
X-Powered-By: PHP/7.4.8
Set-Cookie: PHPSESSID=0kgaorm8uidvc488une11hlmau; path=/; HttpOnly
Referrer-Policy: same-origin
Date: Wed, 09 Jun 2021 06:55:44 GMT
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=UTF-8

<br /> <b>Warning</b>: include once(../connection/dbconn.php): failed to open stream: No such file
or directory in <b >C:\ xampp\htdocs\kswcf\jscript\getdist.php</b> on line <b>3</b><br /> <br />
<b>Warning</b>: include once(): Failed opening '../connection/dbconn.php' for inclusion
(include path= 'C:\ xampp\php\PEAR') in <b >C:\ xampp\htdocs\kswcf\jscript\getdist.php</b> on line
<b>3</b><br /> <br /> <b>Warning</b>: include(../class/Scholar.class.php): failed to open stream:
No such file or directory in <b >C:\ xampp\htdocs\kswcf\jscript\getdist.php</b> on line
<b>4</b><br /> <br /> <b>Warning</b>: include(): Failed opening '../class/Scholar.class.php' for
inclusion (include path= 'C:\ xampp\php\PEAR') in <b >C:\
xampp\htdocs\kswcf\jscript\getdist.php</b> on line <b>4</b><br /> <br /> <b>Fatal error</b>:
Uncaught Error: Class 'Scholar' not found in C:\ xampp\htdocs\kswcf\jscript\getdist.php:5 Stack
trace: #0 {main} thrown in <b >C:\ xampp\htdocs\kswcf\jscript\getdist.php</b> on line <b>5</b><br
/> <br /> <b>Warning</b>: include once(../connection/dbconn.php): failed to open stream: No such
file or directory in <b >C:\ xampp\htdocs\kswcf\jscript\getdist.php</b> on line <b>3</b><br /> <br
/> <b>Warning</b>: include once(): Failed opening '../connection/dbconn.php' for inclusion
(include path= 'C:\ xampp\php\PEAR') in <b >C:\ xampp\htdocs\kswcf\jscript\getdist.php</b> on line
<b>3</b><br /> <br /> <b>Warning</b>: include(../class/Scholar.class.php): failed to open stream:
No such file or directory in <b >C:\ xampp\htdocs\kswcf\jscript\getdist.php</b> on line
<b>4</b><br /> <br /> <b>Warning</b>: include(): Failed opening '../class/Scholar.class.php' for
inclusion (include path= 'C:\ xampp\php\PEAR') in <b >C:\
xampp\htdocs\kswcf\jscript\getdist.php</b> on line <b>4</b><br /> <br /> <b>Fatal error</b>:
Uncaught Error: Class 'Scholar' not found in C:\ xampp\htdocs\kswcf\jscript\getdist.php:5 Stack
trace: #0 {main} thrown in <b >C:\ xampp\htdocs\kswcf\jscript\getdist.php</b> on line <b>5</b><br
/>
```

Remediation Tasks by Severity

[High] http://10.162.2.54:81/kswcf/index.php/AdmInst/InstDetailsEntry - 3 issue(s)

Remediation Tasks	Addressed Security Issues
Review possible solutions for hazardous character injection (High) Parameter: insttype_id (Low) Parameter: insttype_id	SQL Injection Database Error Pattern Found

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[High] http://10.162.2.54:81/kswcf/index.php/AdmSearch/ViewRegStudDetls - 3 issue(s)

Remediation Tasks	Addressed Security Issues
Review possible solutions for hazardous character injection (High) Parameter: rentype (Low) Parameter: rentype	SQL Injection Database Error Pattern Found

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptDetlsApprove - 3 issue(s)

Remediation Tasks	Addressed Security Issues
Review possible solutions for hazardous character injection (High) Parameter: freshren (Low) Parameter: freshren	SQL Injection Database Error Pattern Found

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/DeptverifyDetails - 3 issue(s)

Remediation Tasks	Addressed Security Issues
Review possible solutions for hazardous character injection (High) Parameter: freshren (Low) Parameter: freshren	SQL Injection Database Error Pattern Found

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/EditRegtrStudDetls - 3 issue(s)

Remediation Tasks	Addressed Security Issues
Review possible solutions for hazardous character injection (High) Parameter: freshren (Low) Parameter: freshren	SQL Injection Database Error Pattern Found

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetls - 2 issue(s)

Remediation Tasks	Addressed Security Issues
Review possible solutions for hazardous character injection (High) Parameter: fresh (Low) Parameter: fresh	SQL Injection Database Error Pattern Found

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsC - 2 issue(s)

Remediation Tasks	Addressed Security Issues
Review possible solutions for hazardous character injection (High) Parameter: fresh (Low) Parameter: fresh	SQL Injection Database Error Pattern Found

[High] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegStudDetlsClr - 3 issue(s)

Remediation Tasks	Addressed Security Issues
Review possible solutions for hazardous character injection (High) Parameter: fresh (Low) Parameter: fresh	SQL Injection Database Error Pattern Found

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInst/EditInstDetails - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInst/InstitutionView - 2 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

Remediation Tasks	Addressed Security Issues
Avoid storing sensitive session information in permanent cookies (Low) Response Cookie: ci_session	Permanent Cookie Contains Sensitive Session Information

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInst/UpdateInstitutionMaster - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InsCourseInset - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmInstCourse/InstCourseLoad - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/bank_view - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/BankDelte - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/bankDetsEdit - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/BoardView - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/DistrCtDetsUpdte - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/EligBleDetsUpdte - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRt_baNsk - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNerDetsUpdte - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBanNrDets - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery
(Medium) Path	

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtbaNskDetsUpdte - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery
(Medium) Path	

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoardDetsUpdte - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery
(Medium) Path	

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtBoRdDets - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery
(Medium) Path	

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtDistrCtDets - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery
(Medium) Path	

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtEligBleDets - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery
(Medium) Path	

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtReliGnDets - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery
(Medium) Path	

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/InsRtStateDets - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery
(Medium) Path	

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetsEdit - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery
(Medium) Path	

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NewsUpdtDetSView - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationDetsUpdte - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationEdit - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/NotificationView - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsEdit - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetsUpdte - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/ReliGnDetSView - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsEdit - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateDetsUpdte - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/StateView - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/upDat_baNsk - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmMasters/view_bank_afteredit - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/CourseGroupCreate - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/CourseLoad - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/coursemast_view - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_insert - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/coursetype_view - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/delete_coursetype - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/EditCourseMaster - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/EntryCourseMast - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/InsCourseMaster - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/new_entry - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/update_coursetype - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/Admn/view_coursetype_afteredit - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedCancelList - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProceeedGenList - 1

issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery (Medium) Path

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetProcesSeltedStudList - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery (Medium) Path

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/GetsAlotmentStudList - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery (Medium) Path

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmProcssSeltion/studViewProceedings - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery (Medium) Path

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchBankDets - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery (Medium) Path

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchInstView - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery (Medium) Path

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmSearch/searchStudView - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery (Medium) Path

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStatistics/CoursStaticsAdmFresh - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	Cross-Site Request Forgery (Medium) Path

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStatistics/InstStatVew - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStatistics/StudListView - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStatistics/studViewReg - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/InstRegChangeDets - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/RegStudDetlsCancel - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/AdmStudDets/StudregpageEdit - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/main/captcha - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetInstionSMSmail - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/GetstatusSMSmail - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Medium] http://10.162.2.54:81/kswcf/index.php/SendSMSMailDets/SendIndualsmsact - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form (Medium) Path	Cross-Site Request Forgery

[Low] http://10.162.2.54:81/kswcf/index.php/admn/AdmLogDets - 3 issue(s)

Remediation Tasks	Addressed Security Issues
Review possible solutions for hazardous character injection (Low) Path (Low) Parameter: passwd (Low) Parameter: username	Database Error Pattern Found

[Low] http://10.162.2.54:81/kswcf/index.php/Admn/AdmnFrntPge/8fdb7d1450c595cc4ff6ce7676df7d4f3f2b946f - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Remove source code files from your web-server and apply any relevant patches (Low) Path	Web Application Source Code Disclosure Pattern Found

[Information] http://10.162.2.54:81/kswcf/jscript/getdist.php - 1 issue(s)

Remediation Tasks	Addressed Security Issues
Download the relevant security patch for your web server or web application. (Information) Path	Possible Server Path Disclosure Pattern Found

Advisories and Fix Recommendations

SQL Injection

Application

WASC Threat Classification

SQL Injection
http://projects.webappsec.org/SQL-Injection

Security Risks

It is possible to view, modify or delete database entries and tables

Possible Causes

Sanitation of hazardous characters was not performed correctly on user input

Technical Description

The software constructs all or part of an SQL command using externally-influenced input, but it incorrectly neutralizes special elements that could modify the intended SQL command when sent to the database.

Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, and possibly including execution of system commands.

For example, let's say we have an HTML page with a login form, which eventually runs the following SQL query on the database using the user input:

```
SELECT * FROM accounts WHERE username='$user' AND password='$pass'
```

The two variables, $user and $pass, contain the user credentials entered by the user in the login form.
Therefore, if the user has input "jsmith" as the username, and "Demo1234" as the password, the SQL query will look like this:

```
SELECT * FROM accounts WHERE username='jsmith' AND password='Demo1234'
```

But if the user input "'" (a single apostrophe) as the username, and "'" (a single apostrophe) as the password, the SQL query will look like this:

```
SELECT * FROM accounts WHERE username=''' AND password='''
```

This, of course, is a malformed SQL query, and will invoke an error message, which may be returned in the HTTP response.
An error such as this informs the attacker that an SQL Injection has succeeded, which will lead the attacker to attempt further attack vectors.

Sample Exploit:
The following C# code dynamically constructs and executes a SQL query that searches for items matching a specified name. The query restricts the items displayed to those where owner matches the user name of the currently-authenticated user.

```
...
string userName = ctx.getAuthenticatedUserName();
string query = "SELECT * FROM items WHERE owner = "'"
                        + userName + "' AND itemname = '"
                        + ItemName.Text + "'";
sda = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
sda.Fill(dt);
...
```

The query that this code intends to execute follows:

```
SELECT * FROM items WHERE owner =  AND itemname = ;
```

However, because the query is constructed dynamically by concatenating a constant base query string and a user input string, the query only behaves correctly if itemName does not contain a single-quote character. If an attacker with the user name wiley enters the string "name' OR 'a'='a" for itemName, then the query becomes the following:

```
SELECT * FROM items WHERE owner = 'wiley' AND itemname = 'name' OR 'a'='a';
```

The addition of the OR 'a'='a' condition causes the where clause to always evaluate to true, so the query becomes logically equivalent to the much simpler query:

```
SELECT * FROM items;
```

Fix Recommendation - General

There are several mitigation techniques:
[1] Strategy: Libraries or Frameworks
Use a vetted library or framework that does not allow this weakness to occur, or provides constructs that make it easier to avoid.

[2] Strategy: Parameterization
If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated.

[3] Strategy: Environment Hardening
Run your code using the lowest privileges that are required to accomplish the necessary tasks.

[4] Strategy: Output Encoding
If you need to use dynamically-generated query strings or commands in spite of the risk, properly quote arguments and escape any special characters within those arguments.

[5] Strategy: Input Validation
Assume all input is malicious. Use an "accept known good" input validation strategy: a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on blacklisting malicious or malformed inputs. However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.

Fix Recommendation - PHP

** Filter User Input

Before passing any data to a SQL query, it should always be properly filtered with whitelisting techniques. This cannot be over-emphasized. Filtering user input will correct many injection flaws before they arrive at the database.

** Quote User Input

Regardless of data type, it is always a good idea to place single quotes around all user data if this is permitted by the database. MySQL allows this formatting technique.

** Escape the Data Values

If you're using MySQL 4.3.0 or newer, you should escape all strings with mysql_real_escape_string(). If you are using an older version of MySQL, you should use the mysql_escape_string() function. If you are not using MySQL, you might choose to use the specific escaping function for your particular database. If you are not aware of an escaping function, you might choose to utilize a more generic escaping function such as addslashes().

If you're using the PEAR DB database abstraction layer, you can use the DB::quote() method or use a query placeholder like ?, which automatically escapes the value that replaces the placeholder.

REFERENCES
http://ca3.php.net/mysql_real_escape_string
http://ca.php.net/mysql_escape_string
http://ca.php.net/addslashes
http://pear.php.net/package-info.php?package=DB

** Input Data Validation:

While data validations may be provided as a user convenience on the client-tier, data validation must always be performed on the server-tier. Client-side validations are inherently insecure because they can be easily bypassed, e.g. by disabling Javascript.

A good design usually requires the web application framework to provide server-side utility routines to validate the following:
[1] Required field
[2] Field data type (all HTTP request parameters are Strings by default)
[3] Field length
[4] Field range
[5] Field options
[6] Field pattern
[7] Cookie values
[8] HTTP Response

A good practice is to implement a function or functions that validates each application parameter. The following sections describe some example checking.

[1] Required field
Always check that the field is not null and its length is greater than zero, excluding leading and trailing white spaces.

Example of how to validate required fields:

```
  // PHP example to validate required fields
input) {
      ...
ass = false;
input))>0){
ass = true;
      }
ass;
      ...
  }
  ...
fieldName)) {
      // fieldName is valid, continue processing request
      ...
  }
```

[2] Field data type
In web applications, input parameters are poorly typed. For example, all HTTP request parameters or cookie values are of type String. The developer is responsible for verifying the input is of the correct data type.

[3] Field length
Always ensure that the input parameter (whether HTTP request parameter or cookie value) is bounded by a minimum length and/or a

maximum length.

[4] Field range
Always ensure that the input parameter is within a range as defined by the functional requirements.

[5] Field options
Often, the web application presents the user with a set of options to choose from, e.g. using the SELECT HTML tag, but fails to perform server-side validation to ensure that the selected value is one of the allowed options. Remember that a malicious user can easily modify any option value. Always validate the selected user value against the allowed options as defined by the functional requirements.

[6] Field pattern
Always check that user input matches a pattern as defined by the functionality requirements. For example, if the userName field should only allow alpha-numeric characters, case insensitive, then use the following regular expression:
^[a-zA-Z0-9]+$

[7] Cookie value
The same validation rules (described above) apply to cookie values depending on the application requirements, e.g. validate a required value, validate length, etc.

[8] HTTP Response

[8-1] Filter user input
To guard the application against cross-site scripting, the developer should sanitize HTML by converting sensitive characters to their corresponding character entities. These are the HTML sensitive characters:
< > " ' % ;) (& +

PHP includes some automatic sanitization utility functions, such as htmlentities():

```
  $input = htmlentities($input, ENT QUOTES, UTF-8);
```

In addition, in order to avoid UTF-7 variants of Cross-site Scripting, you should explicitly define the Content-Type header of the response, for example:

```
  <?php

  header('Content-Type: text/html; charset=UTF-8');

  ?>
```

[8-2] Secure the cookie

When storing sensitive data in a cookie and transporting it over SSL, make sure that you first set the secure flag of the cookie in the HTTP response. This will instruct the browser to only use that cookie over SSL connections.

You can use the following code example, for securing the cookie:

```
  <$php

value = "some_value";
time = time()+3600;
ath = "/application/";
domain = ".example.com";
secure = 1;

secure, TRUE);
  ?>
```

In addition, we recommend that you use the HttpOnly flag. When the HttpOnly flag is set to TRUE the cookie will be made accessible only through the HTTP protocol. This means that the cookie won't be accessible by scripting languages, such as JavaScript. This setting can effectively help to reduce identity theft through XSS attacks (although it is not supported by all browsers).

The HttpOnly flag was Added in PHP 5.2.0.

REFERENCES

[1] Mitigating Cross-site Scripting With HTTP-only Cookies:
http://msdn2.microsoft.com/en-us/library/ms533046.aspx
[2] PHP Security Consortium:
http://phpsec.org/
[3] PHP & Web Application Security Blog (Chris Shiflett):
http://shiflett.org/

References and Relevant Links

"Web Application Disassembly with ODBC Error Messages" (By David Litchfield)

Cross-Site Request Forgery

Application

WASC Threat Classification

Cross-site Request Forgery
http://projects.webappsec.org/Cross-Site-Request-Forgery

Security Risks

It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Possible Causes

Insufficient authentication method was used by the application

Technical Description

Even well-formed, valid, consistent requests may have been sent without the user's knowledge. Web applications should therefore examine all requests for signs that they are not legitimate. The result of this test indicates that the application being scanned does not do this.

The severity of this vulnerability depends on the functionality of the affected application. For example, a CSRF attack on a search page is less severe than a CSRF attack on a money-transfer or profile-update page.

When a web server is designed to receive a request from a client without any mechanism for verifying that it was intentionally sent, then it might be possible for an attacker to trick a client into making an unintentional request to the web server which will be treated as an authentic request. This can be done via a URL, image load, XMLHttpRequest, etc., and can result in exposure of data or unintended code execution.
If the user is currently logged-in to the victim site, the request will automatically use the user's credentials including session cookies, IP address, and other browser authentication methods. Using this method, the attacker forges the victim's identity and submits actions on his or her behalf.

Fix Recommendation - General

There are several mitigation techniques:
[1] Strategy: Libraries or Frameworks
Use a vetted library or framework that does not allow this weakness, or provides constructs that make it easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard -
https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
Another example is the ESAPI Session Management control, which includes a component for CSRF -
https://owasp.org/www-project-enterprise-security-api

[2] Ensure that your application is free of cross-site scripting issues (CWE-79), because most CSRF defenses can be bypassed using attacker-controlled script.

[3] Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330) -
http://www.cgisecurity.com/articles/csrf-faq.shtml
Note that this can be bypassed using XSS (CWE-79).

[4] Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.

Note that this can be bypassed using XSS (CWE-79).

[5] Use the "double-submitted cookie" method as described by Felten and Zeller:
When a user visits a site, the site should generate a pseudorandom value and set it as a cookie on the user's machine. The site should require every form submission to include this value as both a form and a cookie value. When a POST request is sent to the site, the request should only be considered valid if the form and cookie values are the same.
Because of same-origin policy, an attacker cannot read or modify the value stored in the cookie. To successfully submit a form on behalf of the user, the attacker would have to correctly guess the pseudorandom value. If the pseudorandom value is cryptographically strong, this will be prohibitively difficult.
This technique requires Javascript, so it may not work for browsers that have Javascript disabled -
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.147.1445

Note that this can probably be bypassed using XSS (CWE-79), or when using web technologies that enable the attacker to read raw headers from HTTP requests.

[6] Do not use the GET method for any request that triggers a state change.

[7] Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.

Note that this can be bypassed using XSS (CWE-79). An attacker could use XSS to generate a spoofed Referer, or to generate a malicious request from a page whose Referer would be allowed.

References and Relevant Links

Cross-site request forgery wiki page
"JavaScript Hijacking" by Fortify

Database Error Pattern Found

Application

WASC Threat Classification

SQL Injection
http://projects.webappsec.org/SQL-Injection

Security Risks

It is possible to view, modify or delete database entries and tables

Possible Causes

Sanitation of hazardous characters was not performed correctly on user input

Technical Description

AppScan discovered Database Errors in the test response, that may have been triggered by an attack other than SQL Injection.
It is possible, though not certain, that this error indicates a possible SQL Injection vulnerability in the application.
If it does, please read the following SQL Injection advisory carefully.

The software constructs all or part of an SQL command using externally-influenced input, but it incorrectly neutralizes special elements that could modify the intended SQL command when sent to the database.

Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks, or to insert additional statements that modify the back-end database, and possibly including execution of system commands.

For example, let's say we have an HTML page with a login form, which eventually runs the following SQL query on the database using the user input:

```
SELECT * FROM accounts WHERE username='$user' AND password='$pass'
```

The two variables, $user and $pass, contain the user credentials entered by the user in the login form.
Therefore, if the user has input "jsmith" as the username, and "Demo1234" as the password, the SQL query will look like this:

```
SELECT * FROM accounts WHERE username='jsmith' AND password='Demo1234'
```

But if the user input "'" (a single apostrophe) as the username, and "'" (a single apostrophe) as the password, the SQL query will look like this:

```
SELECT * FROM accounts WHERE username=''' AND password='''
```

This, of course, is a malformed SQL query, and will invoke an error message, which may be returned in the HTTP response.
An error such as this informs the attacker that an SQL Injection has succeeded, which will lead the attacker to attempt further attack vectors.

Sample Exploit:
The following C# code dynamically constructs and executes a SQL query that searches for items matching a specified name. The query restricts the items displayed to those where owner matches the user name of the currently-authenticated user.

```
...
string userName = ctx.getAuthenticatedUserName();
string query = "SELECT * FROM items WHERE owner = "'"
                         + userName + "' AND itemname = '"
                         + ItemName.Text + "'";
sda = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
sda.Fill(dt);
...
```

The query that this code intends to execute follows:

```
SELECT * FROM items WHERE owner =  AND itemname = ;
```

However, because the query is constructed dynamically by concatenating a constant base query string and a user input string, the query only behaves correctly if itemName does not contain a single-quote character. If an attacker with the user name wiley enters the string "name' OR 'a'='a" for itemName, then the query becomes the following:

```
SELECT * FROM items WHERE owner = 'wiley' AND itemname = 'name' OR 'a'='a';
```

The addition of the OR 'a'='a' condition causes the where clause to always evaluate to true, so the query becomes logically equivalent to the much simpler query:

```
SELECT * FROM items;
```

Fix Recommendation - General

There are several mitigation techniques:
[1] Strategy: Libraries or Frameworks
Use a vetted library or framework that does not allow this weakness to occur, or provides constructs that make it easier to avoid.

[2] Strategy: Parameterization
If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the developer to provide this capability at every point where output is generated.

[3] Strategy: Environment Hardening
Run your code using the lowest privileges that are required to accomplish the necessary tasks.

[4] Strategy: Output Encoding
If you need to use dynamically-generated query strings or commands in spite of the risk, properly quote arguments and escape any special characters within those arguments.

[5] Strategy: Input Validation
Assume all input is malicious. Use an "accept known good" input validation strategy: a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on blacklisting malicious or malformed inputs. However, blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.

Fix Recommendation - PHP

** Filter User Input

Before passing any data to a SQL query, it should always be properly filtered with whitelisting techniques. This cannot be over-emphasized. Filtering user input will correct many injection flaws before they arrive at the database.

** Quote User Input

Regardless of data type, it is always a good idea to place single quotes around all user data if this is permitted by the database. MySQL allows this formatting technique.

** Escape the Data Values

If you're using MySQL 4.3.0 or newer, you should escape all strings with mysql_real_escape_string(). If you are using an older version of MySQL, you should use the mysql_escape_string() function. If you are not using MySQL, you might choose to use the specific escaping function for your particular database. If you are not aware of an escaping function, you might choose to utilize a more generic escaping function such as addslashes().

If you're using the PEAR DB database abstraction layer, you can use the DB::quote() method or use a query placeholder like ?, which automatically escapes the value that replaces the placeholder.

REFERENCES
http://ca3.php.net/mysql_real_escape_string
http://ca.php.net/mysql_escape_string
http://ca.php.net/addslashes
http://pear.php.net/package-info.php?package=DB

** Input Data Validation:

While data validations may be provided as a user convenience on the client-tier, data validation must always be performed on the server-tier. Client-side validations are inherently insecure because they can be easily bypassed, e.g. by disabling Javascript.

A good design usually requires the web application framework to provide server-side utility routines to validate the following:
[1] Required field
[2] Field data type (all HTTP request parameters are Strings by default)
[3] Field length
[4] Field range
[5] Field options
[6] Field pattern
[7] Cookie values
[8] HTTP Response

A good practice is to implement a function or functions that validates each application parameter. The following sections describe some example checking.

[1] Required field
Always check that the field is not null and its length is greater than zero, excluding leading and trailing white spaces.

Example of how to validate required fields:

```
  // PHP example to validate required fields
input) {
      ...
ass = false;
input))>0){
ass = true;
      }
ass;
      ...
  }
  ...
fieldName)) {
      // fieldName is valid, continue processing request
      ...
  }
```

[2] Field data type
In web applications, input parameters are poorly typed. For example, all HTTP request parameters or cookie values are of type String. The developer is responsible for verifying the input is of the correct data type.

[3] Field length
Always ensure that the input parameter (whether HTTP request parameter or cookie value) is bounded by a minimum length and/or a maximum length.

[4] Field range

Always ensure that the input parameter is within a range as defined by the functional requirements.

[5] Field options
Often, the web application presents the user with a set of options to choose from, e.g. using the SELECT HTML tag, but fails to perform server-side validation to ensure that the selected value is one of the allowed options. Remember that a malicious user can easily modify any option value. Always validate the selected user value against the allowed options as defined by the functional requirements.

[6] Field pattern
Always check that user input matches a pattern as defined by the functionality requirements. For example, if the userName field should only allow alpha-numeric characters, case insensitive, then use the following regular expression:
^[a-zA-Z0-9]+$

[7] Cookie value
The same validation rules (described above) apply to cookie values depending on the application requirements, e.g. validate a required value, validate length, etc.

[8] HTTP Response

[8-1] Filter user input
To guard the application against cross-site scripting, the developer should sanitize HTML by converting sensitive characters to their corresponding character entities. These are the HTML sensitive characters:
< > " ' % ;) (& +

PHP includes some automatic sanitization utility functions, such as htmlentities():

```
  $input = htmlentities($input, ENT QUOTES, 'UTF-8');
```

In addition, in order to avoid UTF-7 variants of Cross-site Scripting, you should explicitly define the Content-Type header of the response, for example:

```
  <?php

  header('Content-Type: text/html; charset=UTF-8');

  ?>
```

[8-2] Secure the cookie

When storing sensitive data in a cookie and transporting it over SSL, make sure that you first set the secure flag of the cookie in the HTTP response. This will instruct the browser to only use that cookie over SSL connections.

You can use the following code example, for securing the cookie:

```
  <$php

value = "some_value";
time = time()+3600;
ath = "/application/";
domain = ".example.com";
secure = 1;

secure, TRUE);
  ?>
```

In addition, we recommend that you use the HttpOnly flag. When the HttpOnly flag is set to TRUE the cookie will be made accessible only through the HTTP protocol. This means that the cookie won't be accessible by scripting languages, such as JavaScript. This setting can effectively help to reduce identity theft through XSS attacks (although it is not supported by all browsers).

The HttpOnly flag was Added in PHP 5.2.0.

REFERENCES

[1] Mitigating Cross-site Scripting With HTTP-only Cookies:
http://msdn2.microsoft.com/en-us/library/ms533046.aspx
[2] PHP Security Consortium:
http://phpsec.org/
[3] PHP & Web Application Security Blog (Chris Shiflett):
http://shiflett.org/

References and Relevant Links

"Web Application Disassembly with ODBC Error Messages" (By David Litchfield)

Web Application Source Code Disclosure Pattern Found

Application

WASC Threat Classification

Information Leakage
http://projects.webappsec.org/Information-Leakage

Security Risks

It is possible to retrieve the source code of server-side scripts, which may expose the application logic and other sensitive information such as usernames and passwords

Possible Causes

Debugging information was left by the programmer in web pages
Latest patches or hotfixes for 3rd. party products were not installed
Temporary files were left in production environment

Technical Description

AppScan detected a response containing fragments of application source code.
Application source code should not be accessible to web users, as it may contain sensitive application information and back-end logic.
While such leakage does not necessarily represent a breach in security, it can give an attacker useful guidance for future exploitation.
Leakage of sensitive information may carry various levels of risk and should be limited whenever possible.

Fix Recommendation - General

There are many ways a web application can be coaxed into revealing application source code.
To ensure that your application does not allow web users access to source code:
[1] Check that all system patches related to source code disclosure are installed.
[2] Check that no application source code is left in HTML comments.
[3] Check that all source code files are removed from the production environment.

References and Relevant Links

Permanent Cookie Contains Sensitive Session Information

Application

WASC Threat Classification

Insufficient Session Expiration
http://projects.webappsec.org/Insufficient-Session-Expiration

Security Risks

It may be possible to steal session information (cookies) that was kept on disk as permanent cookies

Possible Causes

The web application stores sensitive session information in a permanent cookie (on disk)

Technical Description

During the application test, it was detected that sensitive session information such as user credentials or session tokens were stored in a permanent cookie on the client's computer.

[1] Since other users may use the computer, this information may be compromised or used for identity theft or user impersonation.

[2] If the computer will be compromised, the account information may be stolen and used later by a malicious user.

In addition, several privacy regulations require that users will be identified uniquely before accessing sensitive information. Since a permanent cookie may allow other users to logon to the web application without authenticating, this may not comply with several privacy regulations.

Fix Recommendation - General

Make sure that sensitive session information such as user credentials or session tokens will always be stored in non-permanent cookies (RAM cookies) only. This is achieved by not setting the "Expires" field in the cookie.

References and Relevant Links

Financial Privacy: The Gramm-Leach Bliley Act
Health Insurance Portability and Accountability Act (HIPAA)
Sarbanes-Oxley Act
California SB1386
HTTP State Management Mechanism (RFC 2109)

Possible Server Path Disclosure Pattern Found

Application

WASC Threat Classification

Information Leakage
http://projects.webappsec.org/Information-Leakage

Security Risks

It is possible to retrieve the absolute path of the web server installation, which might help an attacker to develop further attacks and to gain information about the file system structure of the web application

Possible Causes

Latest patches or hotfixes for 3rd. party products were not installed

Technical Description

AppScan detected a response containing a file's absolute path (e.g. c:\dir\file in Windows, or /dir/file in Unix).

An attacker may be able to exploit this information to access sensitive information on the directory structure of the server machine which could be used for further attacks against the site.

Fix Recommendation - General

There are several mitigation techniques:
[1] In case the vulnerability is in the application itself, fix the server code so it doesn’t include file locations in any output.
[2] Otherwise, if the application is in a 3rd party product, download the relevant security patch depending on the 3rd party product you are using on your web server or web application.

References and Relevant Links

HCL customers are responsible for ensuring their own compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws.